Evaluation of Performance Tools on Peta-Scale Systems for CQoS Research

Motivation

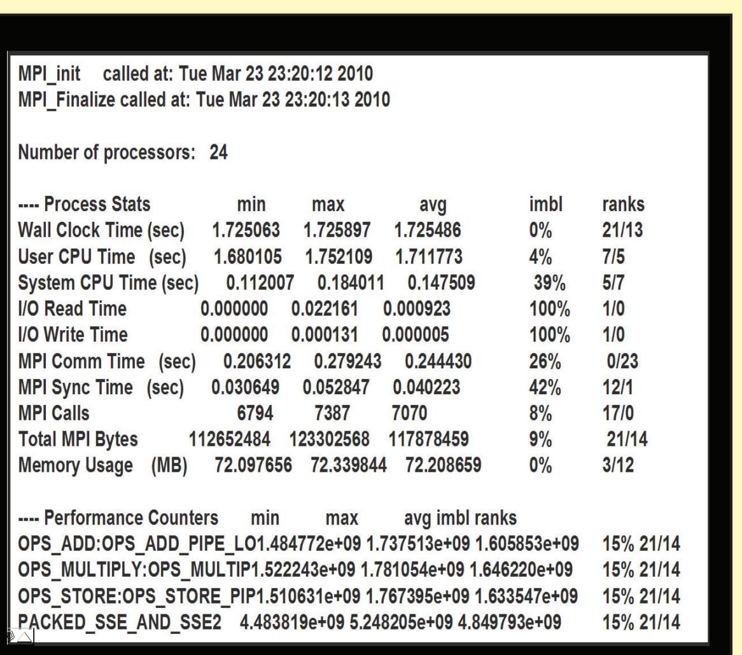
When conducting performance research such as optimization, automatic tuning, adaptive computing or computational quality of service(CQoS), it is necessary to acquire performance data of proper granularity for analyses. While a basic timer gives the overall wall clock time, on complex peta-scale supercomputers we may need to use performance tools to gain insight into performance data of different groups, such as MPI, IO, cache utilization, etc.. There are several performance tools available on Cray-XT5 systems, and each performance tool provides different amounts of data, and has different complexity of usage (from a few options to a lot of options). It may take a few days just to learn some options provided by a tool. The goal of this project is to investigate the strength of each performance tool and provides scientists an understanding of when to use what in performance research.

Approaches and Research Results

Four major performance tools on NSF's Kraken are investigated: FPMPI, IPM, CrayPAT and TAU. For each tool we evaluated the amount of data/information it generates, the learning curve, the overhead it may incur and the suitability of use for automatic tuning/computational quality of service research. GAMESS and other applications or benchmarking suits, such as DNS code or HPC Challenges, are used to conduct the investigation.

FPMPI

- Very easy to use – just link it!!!
- It provides text output only.
- Performance data are divided into groups such as MPI, IO, wall-clock time.
- It shows performance data of MPI in details.
- A good tool to gain an ***overview*** of application performance.

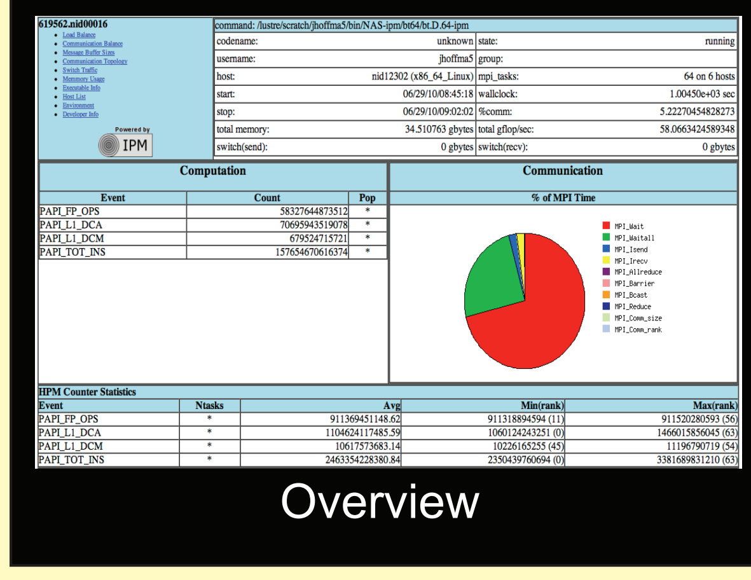


```
MPI_init     called at: Tue Mar 23 23:20:12 2010
MPI_Finalize called at: Tue Mar 23 23:20:13 2010

Number of processors:   24

---- Process Stats          min       max       avg        imbl    ranks
Wall Clock Time (sec)    1.725063  1.725897  1.725486      0%     21/13
User CPU Time   (sec)    1.680105  1.752109  1.711773      4%     7/5
System CPU Time (sec)    0.112007  0.184011  0.147509      39%    5/7
I/O Read Time            0.000000  0.022161  0.000923      100%   1/0
I/O Write Time           0.000000  0.000131  0.000005      100%   1/0
MPI Comm Time   (sec)    0.206312  0.279243  0.244430      26%    0/23
MPI Sync Time   (sec)    0.030649  0.052847  0.040223      42%    12/1
MPI Calls                    6794      7387      7070      8%     17/0
Total MPI Bytes         112652484 123302568 117878459      9%     21/14
Memory Usage    (MB)    72.097656 72.339844 72.208659      0%     3/12

---- Performance Counters      min       max        avg imbl ranks
OPS_ADD:OPS_ADD_PIPE_LO1.484772e+09 1.737513e+09 1.605853e+09   15% 21/14
OPS_MULTIPLY:OPS_MULTIP1.522243e+09 1.781054e+09 1.646220e+09   15% 21/14
OPS_STORE:OPS_STORE_PIP1.510631e+09 1.767395e+09 1.633547e+09   15% 21/14
PACKED_SSE_AND_SSE2   4.483819e+09 5.248205e+09 4.849793e+09    15% 21/14
```

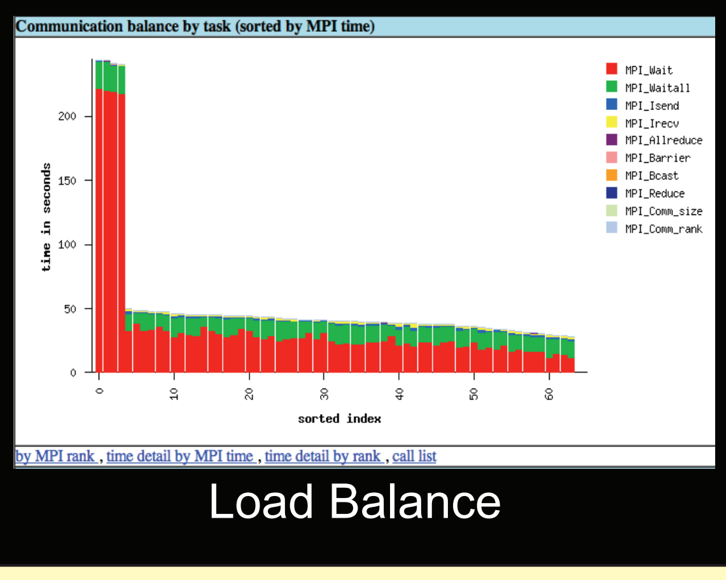
Text output from FPMPI. Data are divided into groups. Hardware counters information can be acquired through PAPI.

IPM

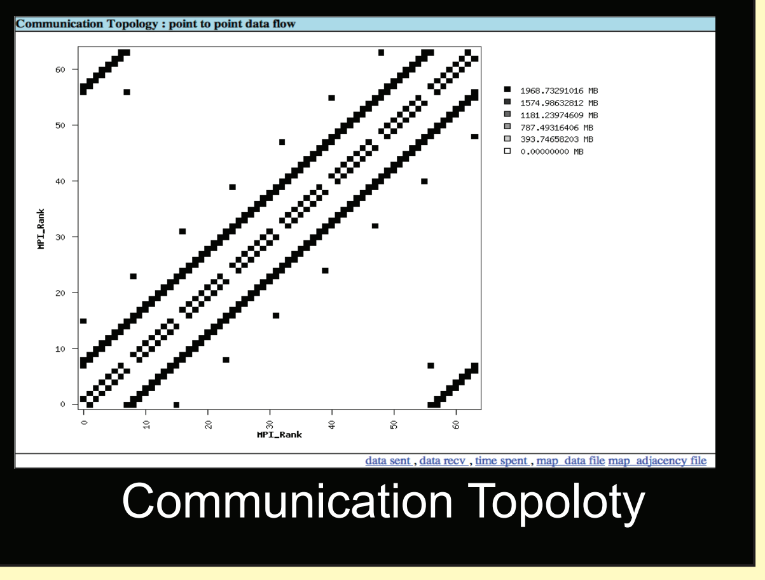
- Easy to use (just like it too) if you are using text output only.
- It provides graphical output, but needs a html viewer to view the data. Procedure to generate graphical data is not intuitive.
- Besides basic profiling data, IPM provides more detailed performance data such as communication topology, volume, load balance.



Overview



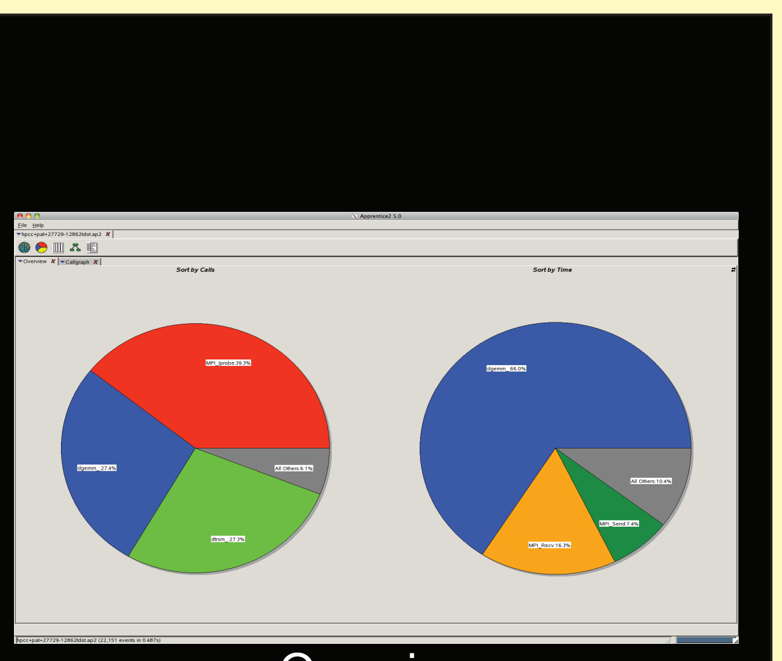
Load Balance



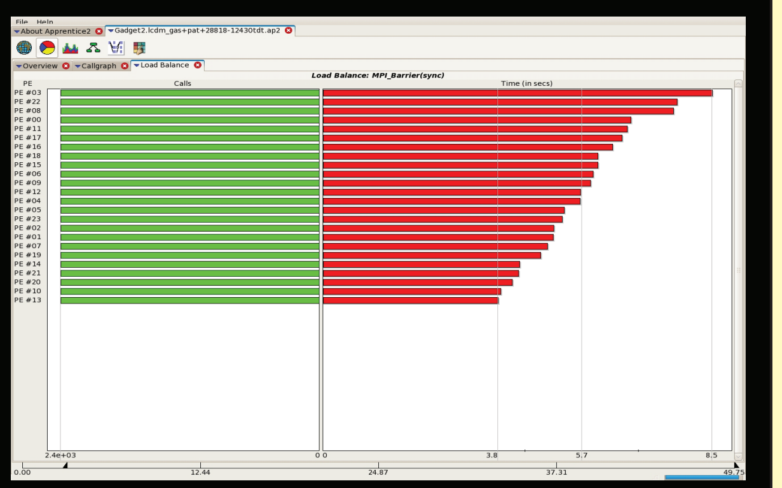
Communication Topoloty

CrayPAT

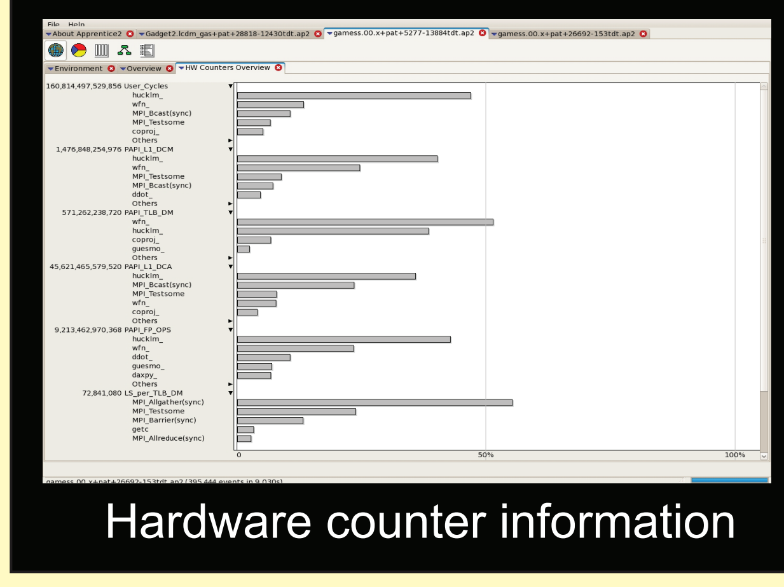
- A set of tools: pat_build for instrumentation, pat_report for post-processing and apprentice2 for viewing graphical output.
- It has many automatic analysis features, and provides basic & detailed profiling data and has tracing capability.
- Modest learning curve.
- A very versatile and easy to use tool to analyze performance data from *a single run* on Cray platforms.



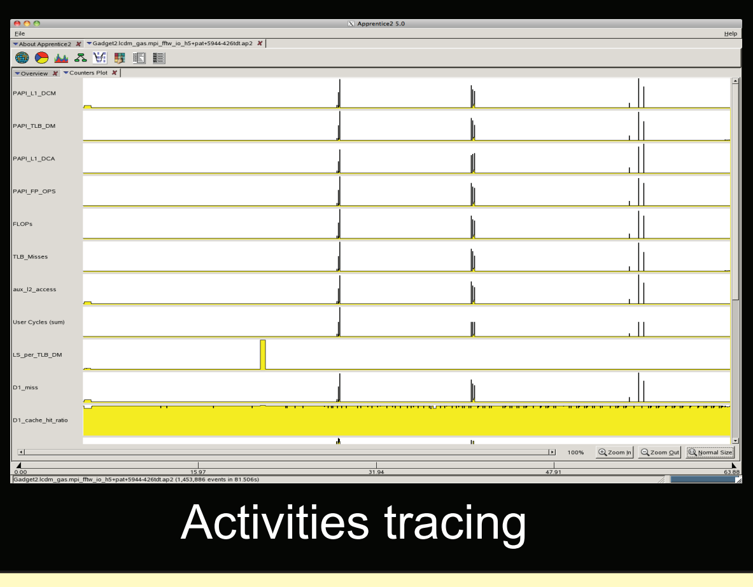
Overview



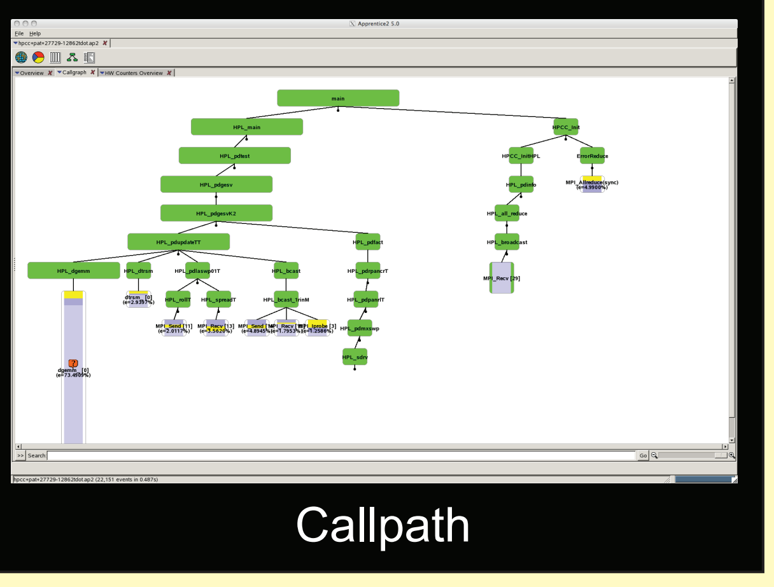
Load Balance



Hardware counter information



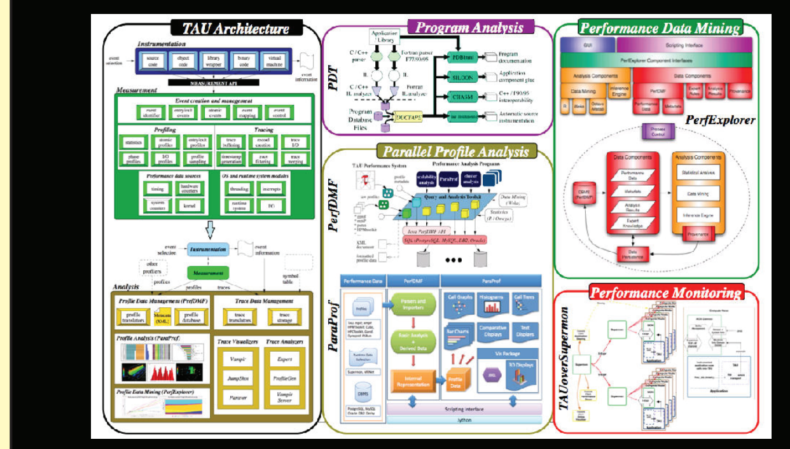
Activities tracing



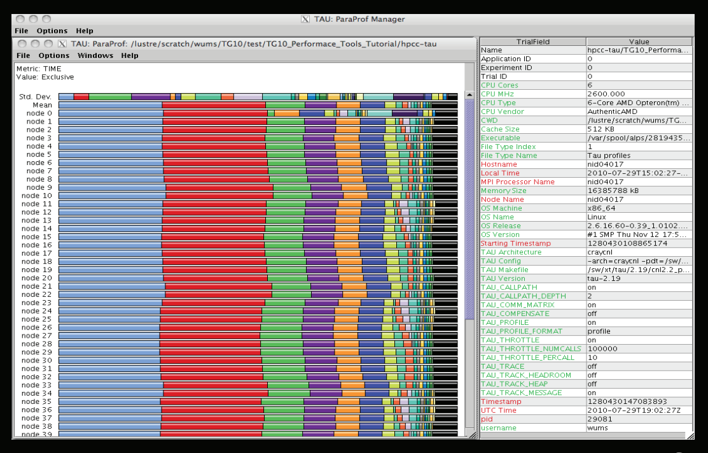
Callpath

TAU

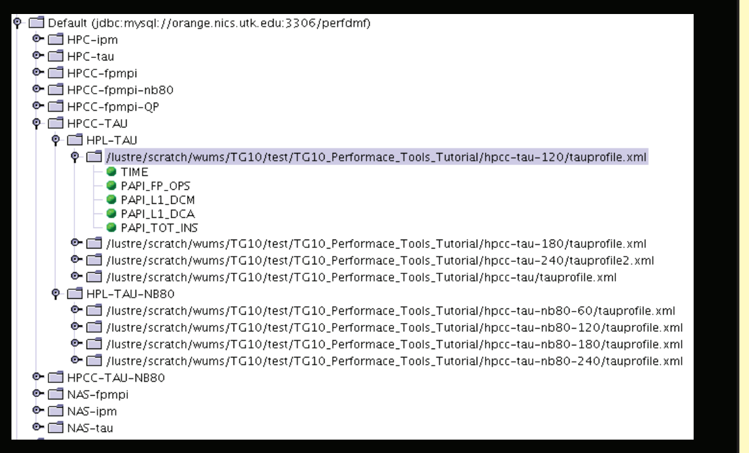
- A system – it can analyze data from ***many runs***!
- It provides both text & graphical output.
- It provides many manual instrumentation options
- Through PerfDMF/PerfExplorer, TAU can link to a database, external tools such as R to conduct complex analysis for large amounts of performance data.
- ***Steep learning curve***!!!



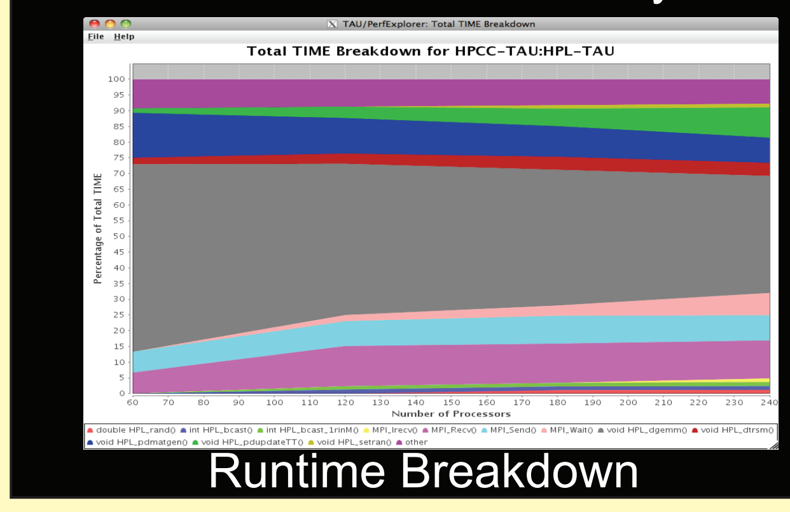
The structure of the TAU System



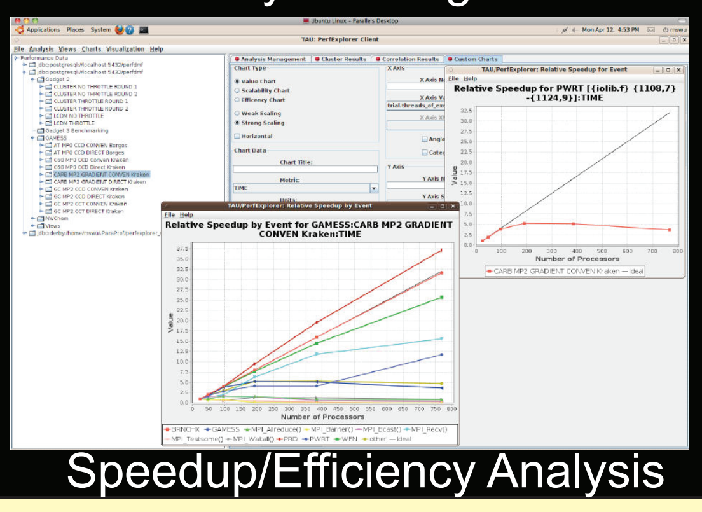
Data Analysis using ParaProf



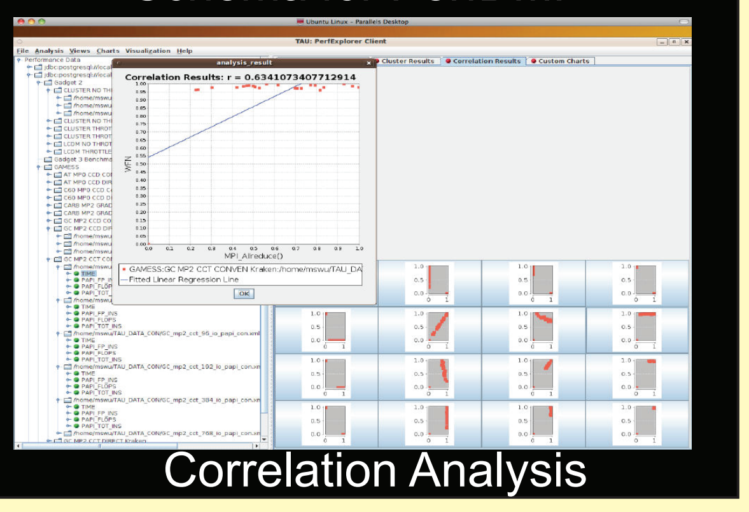
Schema for PerfDMF



Runtime Breakdown



Speedup/Efficiency Analysis



Correlation Analysis

Discussion and Conclusions

The complexity of usage increase roughly from FPMPI to IPM to CrayPAT to TAU. For CQoS or automatic tuning research, a portable performance tool that can acquire performance data of the same granularity on different platforms is essential. While vendor proprietary tools on their own machines provide many useful features and are easy to use, lack of portability makes these tools difficult to use for CQoS research. On the other hand, FPMPI and IPM, while easy to use and portable, may not provide enough information or do not have the capability to handle large amounts of performance data; currently only the TAU system provides such a capability, although limite.

Automatic tuning research requires analyzing/connecting application metadata with performance data. While this can be done through PerfDMF/PerfExplorer, many laborious efforts have to be invested. The trend of performance tool development is still focused on "analyzing a single run", but the 'snapshot' information is usually not enough for CQoS research. More support from the performance tool development community to process large amounts of performance data can be very helpful for CQoS/automatic tuning research.

Contact Information: Meng-Shiou Wu mwu5@utk.edu, mswu@scl.ameslab.gov